

Tecnología Dockers: ¿Qué es?, sus ventajas y beneficios como herramientas para pruebas de aplicaciones

Dockers technology: What is it? Its advantages and benefits as a tool for application testing.

Carlos Rolando Jacinto Rivera^{1,2},

1 Facultad Multidisciplinaria Paracentral, Universidad de El Salvador, El Salvador

2 ORCID: 0009-0004-0537-6813

Correspondencia

jr10013@ues.edu.sv

RESUMEN

La presente investigación se desarrolló por la necesidad profundizar en la temática de “Dockers”, ya que esta tecnología permite la implementación y desarrollo de sistemas o aplicativos informático, ejecutándose en un ambiente aislado, y administrado con una gran facilidad, asimismo, teniendo la cualificación de soportar la implementación en diferentes sistemas operativos en la que se ejecute dicha tecnología, puesto que el software dentro del contenedor ejecutará todas sus tareas en su entorno nativo, al mismo tiempo, brinda diferentes herramientas para la modificación, actualización y masificar los distintos contenedores.

Palabras clave: Docker, Funciones, Configuraciones, Dockerfile, Contenedores

ABSTRACT

This research was developed due to the need to delve deeper into the subject of “Dockers,” as this technology allows for the implementation and development of computer systems or applications, running in an isolated environment, and easily managed. It also has the ability to support implementation on different operating systems on which this technology runs, since the software inside the container will execute all its tasks in its native environment, while also providing different tools for modifying, updating, and massifying the different containers.

Keywords: Docker, Functions, Configurations, Dockerfile, Containers.

INTRODUCCIÓN

La Universidad de El Salvador (UES) fue fundada el 16 de febrero de 1841 por medio de un decreto emitido por la Asamblea Constituyente, que recién se había instalado, y suscrito por el diputado presidencial Juan José Guzmán y los diputados secretarios Leocadio Romero y Manuel Barberén (UES, 2024), posteriormente frente a las demandas de profesionales en el área de ingeniería se fundó en 1954 el departamento de Ingeniería Industrial de la Universidad de El Salvador, luego para 1966 se aprobaron los planes de estudio en la Facultad de Ingeniería y Arquitectura, de la Universidad de El Salvador, y para 1991 existían dos departamentos, entre los cuales es preciso mencionar la creación del departamento en sistemas, siendo esta que se consolidó tanto, al grado que la Escuela de Ingeniería Industrial absorbía alrededor del 40% de la población de la facultad, y dicho acontecimientos más relevante fue el inicio de las gestiones para la separación de las carreras de Ingeniería Industrial e Ingeniería de Sistemas Informáticos (Facultad de Ingeniería y Arquitectura, 2024).

Con el objetivo de formar profesionales con conocimientos científico-tecnológicos en el campo de la informática, comprometido/as a resolver problemas ingenieriles de Sistemas de Información en empresas públicas y privadas, contribuyendo con ello al desarrollo socio-económico de El Salvador (SG-UES, 2024), es indispensable la necesidad de seguir constantemente capacitándose en las novedosas herramientas y tecnologías informáticas, por lo que dicha investigación se centra en la comprensión lógica de la Tecnología Dockers que sirve para la simplificación del proceso de desarrollo de aplicaciones, es decir, es útil al agilizar la gestión de aplicaciones al ejecutar las mismas en contenedores para encapsularlos y ejecutarlos de la manera correcta en cualquier entorno, debido a que proporciona portabilidad, eficiencia y escalabilidad a las aplicaciones.

DESARROLLO

DOCKERS

Los Docker son tecnologías desarrolladas hacia el año 2013 introduciendo un estándar en la industria de la

computación para el uso y desarrollo de contenedores, es decir, esta tecnología proporciona las bases necesarias para ejecutar aplicaciones compuestas en la nube, permitiendo que estos sean administrados desde un solo anfitrión y evolucionando el método de virtualización más liviano para todo el sistema operativo (Yuriana Noreño, 2022); asimismo, al hablar de esta forma de acceder a la tecnología, desarrollar, desplegar y ejecutar las aplicaciones en entornos aislados [denominados, "Contenedores"] que permite empaquetar las aplicaciones desde un solo contenedor mediante un código fuente de la aplicación, los archivos de configuración y todas las dependencias software que necesita.

Esta estrategia permite que las aplicaciones se puedan ejecutar de la misma manera sobre cualquier infraestructura que tenga soporte para Docker, tanto de forma local como en la nube, y denotando que la relevancia por un software queda en segundos planos. Yuriana Noreño (2022) siendo una forma novedosa de trabajar designada como "it works on my machine", haciendo alusión a que la aplicación puede funcionar correctamente en el entorno de desarrollo, pero tiene errores en el entorno de producción, porque los dos entornos no son idénticos y contienen versiones de software diferentes -dependiendo del caso aislado en particular del cual se explique-, (Yuriana Noreño, 2022).

Añadiendo que para la comprensión conceptual sobre la accesibilidad de los Dockers, es importante esclarecer en primer lugar, que una "Máquina Virtual" es una emulación en un sistema real de computación con su propia CPU, memoria, interfaz de red y almacenamiento; estos vienen compuestos por un conjunto de archivos que en su mayoría, se pueden manipular como si se tratara de cualquier otro archivo presente en algún sistema operativo.

Las máquinas virtuales virtualiza la aplicación y todas las librerías y dependencias, incluyendo el sistema operativo lo que demanda mayor tiempo arranque en el servidor en un entorno aislado; asimismo, ese denominado "Entorno Aislado o Contenedor" es un camino diferente de virtualización y en la forma como se construyen las aplicaciones ya que permiten el aislamiento de recursos, virtualizando el sistema operativo y pro-

vocando un menor consumo de recursos para ejecutar una aplicación.

Particularmente las tecnologías Docker conservan un proceso exclusivo para la creación y exposición de imágenes a través de sus repositorios, además, los contenedores Docker tienen la propiedad de ejecutar varias instancias de una imagen permitiendo iniciarlas,

detenerlas o conectarlas a otros contenedores mediante redes. (Yuriana Noreño, 2022) Para este punto, es necesario añadir el siguiente cuadro comparativo entre las cualidades más destacables entre Contenedores y Máquinas, de la siguiente manera:

Las principales ventajas que nos aporta el uso de Docker son las siguientes:

Tabla 1. *Comparative Between MV y Containers.*

Máquina Virtual:	Contenedores
• Necesidad de un sistema operativo para cada servicio • Inicio y apagado lentos • Consume toda la RAM asignada a la MV • Red de transferencia de datos a través de forma de hipervisor. • cuellos de botella • Comparta archivos con el Host a través del Hipervisor. • Directo el acceso no es posible	• Comparte los componentes del sistema del host; no se Necesita un sistema operativo para cada servicio. • Inicio y apagados rápidos • Comparta archivos con el host mediante Copia segura (SCP) • Crea un puente de red para cada servicio que permite utilizar todo el ancho de banda. • Consume solo la RAM necesaria del servicio.

Nota. Datos tomados del documento A performance evaluation between Docker container and Virtual Machines in cloud computing architectures (Luis Herrera 2017)

- ◇ Soluciona el problema «it works on my machine».
- ◇ Permite tener un entorno de desarrollo limpio, seguro y portátil.
- ◇ Facilita la automatización de pruebas, integración y empaquetado.
- ◇ Posibilita empaquetar una aplicación con todas las dependencias que necesita (código fuente, librerías, configuración, etc.) para ser ejecutada en cualquier plataforma.
- ◇ Se eliminan inconsistencias entre los entornos de desarrollo, pruebas y producción. El proceso de despliegue es rápido y repetible.

Instalación y Configuración

En primera instancia es necesario ingresar a sitio web oficial “Docker Desktop N°1” e instalar en la pesta-

ña correspondiente a su ordenador, (Windows, Mac o Linux) con el siguiente Link: <https://www.docker.com/products/docker-desktop/> (Pelado Nerd, 2020); asimismo, para Linux se puede crear un servidor en “Arsys Servidor Cloud” para Docker y hacerse de un dominio público, variando precios de entre 15-230 Euros, convertido sería entre \$16.28 - \$249.58 Dólares. (ARSYS, 2024)

Requisitos para Windows:

- WSL versión 1.1.3.0 o posterior.
- Windows 11 de 64 bits: Home o Pro versión 21H2 o superior, o Enterprise o Education versión 21H2 o superior.
- Windows 10 de 64 bits: Recomendamos Home o Pro 22H2 (compilación 19045) o superior, o Enterprise o Education 22H2 (compilación 19045) o superior, por el contrario el

mínimo requerido es Home o Pro 21H2 (compilación 19044) o superior, o Enterprise o Education 21H2 (compilación 19044) o superior.

- Procesador de 64 bits, Ram 4 GB, y habilitar la virtualización de hardware en BIOS. (Docker. docs, 2024)

Requisitos para MacOS:

- Una versión compatible con MacOS
- Al menos 4 Ram (Docker.Docs, 2024)

Plataformas Compatibles con Linux:

- Ubuntu
- Debian
- Red Hat Enterprise Linux (RHEL)
- sombrero(Docker.docs, 2024)

Configuración:

1. Descargar la aplicación desde el sistema operativo deseado en el siguiente Link: <https://www.docker.com/products/docker-desktop/> (Windows, Mac o Linux)
2. Hacer doble clic en Docker Desktop Installer exe. para ejecutar el instalador, de forma pre-determinada
3. Posteriormente se instalará C:\Program Files\ Docker\Docker
4. seguidamente se presentará un panel con los siguientes pasos para autorizar el instalador y seguir la instalación (según sea su sistema-MacOS, Windows, Linux)
5. Posteriormente sea exitosa su instalación, cierre la pestaña para completar el proceso, y le aparecerá en su ordenador la App. de Docker.

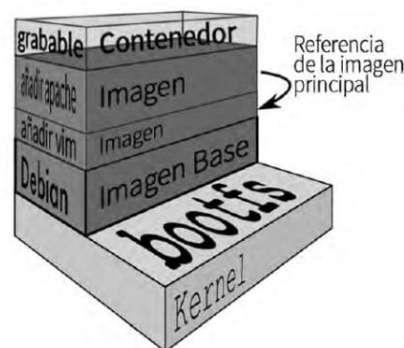
Imágenes Dockers y Dockerfile

Cuando se hace referencia a una imagen en Docker quiere decir un archivo estático, una lectura, es decir, una plantilla asemejándose a una máquina virtual, que si bien es cierto, podría duplicarse las herramientas o funcionalidades de esta plantilla, similar a un "power-

point" pero el material diferenciador sería en cómo se utiliza esta plantilla.(Alberto Lopez-Tech Chip, 2021) básicamente, Una imagen contiene distintas capas de datos (la distribución, diferente software, librerías y la personalización), y para una descripción mental se mostrará la siguiente imagen(1):

Docker nos ofrece un repositorio oficial (Docker

Figura 1. Representación de una Imagen en Docker



Nota. Se sustrajo la imagen del documento Aprender Docker Un Enfoque Práctico (José Juan Sánchez Hernández, 2022)

Hub) donde están disponibles imágenes oficiales para distribuciones (debian, ubuntu, centos, etc.) y para aplicaciones (httpd, nginx, mariadb, etc.). Además, en este repositorio, los usuarios tienen la opción de subir sus propias imágenes que han generado.

La dirección para el repositorio oficial de Docker es la siguiente: <https://hub.docker.com/>. En esta dirección es posible buscar imágenes, ver cómo utilizar la imagen, distinguir las diferentes versiones (tags) y diversa información de gran utilidad.

Las imágenes creadas entre la imagen base y la imagen definitiva se llaman imágenes intermedias. Un ejemplo es una imagen con una distribución que contiene los elementos básicos. Posteriormente, en esa imagen se han instalado algunas utilidades, una aplicación y sus librerías. Para finalizar, se establece un software de monitorización para dicha aplicación.

Posteriormente al identificar que es una Imagen en Docker, es necesario entender que dentro del repositorio oficial se puede buscar una imagen sin tener que abrir el navegador, con los comandos (docker search

búsqueda [opciones]) donde se podrá observar el nombre de la imagen, una breve descripción, los valores STARS, si es oficial o no, y si es una imagen automática. seguidamente para “descargar una imagen” se utiliza el comando `docker pull imagen[:tag]` comprobando si la imagen existe en el repositorio, su versión, actualización o el lastes sino se identifica, descomprimir la imagen descargada, y asimismo, esto incluirá el listado de accesibilidad de la imagen y generará un identificador corto o largo de 64 caracteres basados en el algoritmo `sha256`.

Además de acceder al historial de la imagen mediante los comandos `docker history imagen`, estos mostraran la imagen base dentro del sistema operativo, dos imágenes intermedias (luego de su instalación de utilidades y la instalación de la aplicación y librerías) y una imagen final, siendo esta última la unión del resto, sin embargo, también se añaden los comandos de `docker save` o `docker import` imagen - para generar una copia

de seguridad y por el contrario, generar que la imagen se resetee a su forma original; asimismo, con el comando `docker rmi` imagen, procederá a eliminar la imagen trabajada. (José Hernández, 2022)

Ahora bien, desde otra óptica al hablar de imágenes ya creadas y/o modificadas, también “Docker” ofrece las herramientas para la creación de imágenes mediante un script que guía la manera como se construye la imagen de Docker, esta es usada con una sintaxis específica que contiene todos los comandos necesarios para ensamblar la imagen la cual le explica al Demon Docker cómo construirla. (Yuriana Noreño, 2022)

Este scrip o fichero denominado “Dockerfile” es un documento de texto que contiene todos los comandos que un usuario puede llamar en la línea de comando para ensamblar una imagen, por lo se muestra la siguiente tabla con los formatos a utilizar y una breve descripción de sus funciones dentro de la aplicación.

Tabla 2. Comandos para un Dockerfile.

Comando	Función
ADD	Agregue archivos y directorios locales o remotos.
ARG	Utilice variables de tiempo de construcción.
CMD	Especifique comandos predeterminados.
COPY	Copiar archivos y directorios.
ENTRYPOINT	Especifique el ejecutable predeterminado.
ENV	Establecer variables de entorno.
EXPOSE	Describe en qué puertos escucha tu aplicación.
FROM	Cree una nueva etapa de construcción a partir de una imagen base.
HEALTHCHECK	Verifique el estado de un contenedor al inicio.
LABEL	Agregar metadatos a una imagen.
MAINTAINER	Especifique el autor/a de una imagen.
ONBUILD	Especifique instrucciones para cuando se utiliza la imagen en una compilación.
RUN	Ejecuta comandos de compilación.
SHELL	Establece el shell (estructura) predeterminado de una imagen.
STOPSIGNAL	Especifique la señal de llamada del sistema para salir de un contenedor.
USER	Establecer ID de usuario y grupo.
VOLUME	Crea montajes de volumen.
WORKDIR	Cambiar directorio de trabajo.

Nota. Se adquirió la información del sitio web [Docker.docs](https://docs.docker.com/engine/reference/dockerfile/#/) (Dockerfile Reference 2024)

Formato de Accesibilidad de comandos: #Comando, INSTRUCCIÓN argumento

Dockerfile al darle las instrucciones no distingue entre mayúsculas y minúsculas. Sin embargo, la convención es que estén en MAYÚSCULAS para distinguirlos de los argumentos más fácilmente.

Asimismo, Docker ejecuta instrucciones en un Dockerfile en orden, ya que Dockerfile debe comenzar con una instrucción FROM, dado que ocurrirá después de directivas del analizador, comentarios y ARG de alcance global, también la instrucción FROM especifica la imagen principal a partir de la cual está construyendo, y FROM solo puede ir precedido de una o más instrucciones ARG, que declaran argumentos que se utilizan en las líneas FROM del Dockerfile.

Las directivas del analizador son opcionales y afectan la forma en que se manejan las líneas posteriores en un Dockerfile, dentro del analizador no agregan capas a la compilación y no aparecen como pasos de compilación, y se escriben como un tipo especial de comentario con el formato # directiva=valor, esta única directiva sólo podrá utilizarse una vez. (Dockers.docs, 2024)

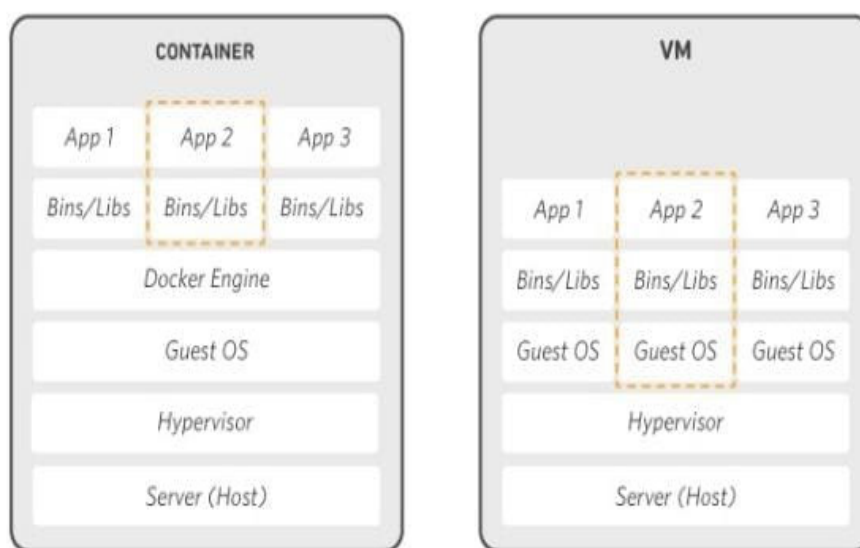
Contenedores Docker

Posteriori al desarrollo conceptual sobre un “Docker” entendiéndose como una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente, es necesario comprender que dicho software empaquetado se encuentra almacenado en una unidad denominada **“Contenedor”**.

Siendo esta la que alberga todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución, pudiéndose implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará en tiempo real.

Los Dockers proporcionan una manera estandarizada de ejecutar un código, siendo su sistema operativo un contenedor, similar aunque no igual que una máquina virtual, sin embargo, los contenedores virtualiza el sistema operativo en un servidor. (ASW, 2024) La colaboración de los docker ofrece a los desarrolladores y administradores una manera muy confiable y económica de crear, enviar y ejecutar aplicaciones distribuidas en cualquier escala, lo que genera la producción sobre la entrega de las aplicaciones en la nube, representado con la siguiente imagen:

Figura 2. Servicios de AWS



Nota. Según AWS Amazon como AWS Fargate, Amazon ECS, Amazon EKS y AWS Batch (AWS, ¿Qué es un Docker? 2024)

Esta forma de almacenamiento (contenedores de producción) cada vez más en auge muestra diferentes tipos de aplicabilidad en la actualidad, mostrando en 2019 el uso de este sistema operativo por el 35% de

empresas; por otro lado, se muestra a continuación una tabla con información de las plataformas más gestionadas en 2022:

Tabla 3. Plataformas Más Gestionadas para Contenedores 2022

Plataformas Gestionadas para Contenedores		
Plataforma	¿A qué se dedica?	Producto y función
Amazon Elastic Container Service (Amazon ECS)	Es un servicio de orquestación de contenedores completamente administrado mediante el cual las empresas implementan, administran y escalan aplicaciones en contenedores.	Amazon Web Services (AWS) , las organizaciones interesadas pueden optimizar su tiempo gracias a la capacidad informática sin servidor para contenedores que les facilita AWS Fargate , es un motor informático que, por sus características, y al permanecer integrado en Amazon ECS, hace que los usuarios no tengan que preocuparse de administrar servidores, manejar la planificación de la capacidad o ahondar en cómo aislar las cargas de trabajo de los contenedores por seguridad la incorporación de la función Amazon Elastic Container Service AnyWhere , su propósito es que las compañías ejecuten y gestionen las cargas de trabajo en una infraestructura administrada por el cliente. Además de la definición de tareas por medio de una plantilla de notación de objetos JavaScript o comúnmente denominados por (JSON) denominada Task Definition (Definición de tarea).
Cisco Inter-sight Kubernetes Service (IKS)	Esta solución "SaaS" llave en mano lleva los proyectos Kubernetes en producción a múltiples nubes. Entre otros, se despliega sobre hipervisores VMware ESXi. Con Kubernetes (K8s), esta plataforma de orquestación de contenedores con la que los equipos de desarrollo despliegan, gestionan y escalan sus aplicaciones en contenedores.	La instalación de Kubernetes y los diferentes componentes de software necesarios, la creación de clústeres, la configuración del almacenamiento, la red y la seguridad, la optimización para IA/ML y otras tareas manuales pueden ralentizar el ritmo de desarrollo y hacer que los equipos pasen horas depurando. Así, IKS, como solución SaaS llave en mano , constituye una plataforma de gestión de contenedores ligera y contrastada para llevar los proyectos Kubernetes en producción a múltiples nubes. Con una instalación que dura minutos, se despliega sobre hipervisores VMware ESXi, hipervisores Cisco HyperFlex Application Platform (HXAP) y/o directamente en servidores bare metal Cisco HyperFlex Application Platform para un importante ahorro y eficiencia sin necesidad de virtualización. Además, con Cisco HXAP aprovechando las capacidades de virtualización nativa de contenedores, se pueden ejecutar máquinas virtuales (VM), contenedores basados en VM y contenedores bare metal en la misma plataforma.
Docker Swarm	Incluye la administración de clústeres integrada con Docker Engine, el balanceo de carga, opciones de seguridad como el cifrado y la autenticación, y actualizaciones continuas. Esta opción engloba un conjunto de productos bajo la 'fórmula' plataforma como servicio que emplea la virtualización a nivel del sistema operativo para entregar software en paquetes llamados contenedores. Se ejecuta en cualquier máquina que incorpore los sistemas operativos Linux, macOS o Windows, Docker facilita que las aplicaciones se ejecuten en una variedad de ubicaciones: instalaciones del cliente, nube pública o nube privada.	Los contenedores están aislados unos de otros y que agrupan su propio software, bibliotecas y archivos de configuración, pudiendo comunicarse entre sí a través de unos canales definidos. Debido a que todos los contenedores comparten los servicios de un único kernel de sistema operativo, utilizan menos recursos que las máquinas virtuales, y además pueden integrarse con algunos servicios de las siguientes plataformas Amazon Web Services, Google Cloud Platform, IBM Bluemix o Microsoft Azure. Docker Swarm , permite que un grupo de motores Docker se convierta en un único motor Docker virtual, y desde la versión Docker 1.12, este modo Swarm se encuentra integrado en el software Docker Engine, añadiendo como ventaja de modificar las configuraciones de los servicios, incluidas las redes y los volúmenes a los que se está conectado, sin necesidad de reiniciarlos manualmente.

Google Kubernetes Engine (GKE)	Sus características incluyen copia de seguridad de GKE, gestión de acceso e identidades, registro y monitorización, autoescalado, limitación de recursos, aislamiento de contenedores, reparación automática, además de administrarlo por la nube de Google, Google Kubernetes Engine (GKE) es un servicio que ayuda a las empresas a desplegar, escalar y gestionar Kubernetes automáticamente. Además que contemplan la migración de cargas de trabajo tradicionales a contenedores de GKE deben centrarse en Migrate for Anthos and GKE	A este respecto, la plataforma sobre la que se asienta -preparada para escalar hasta 15.000 nodos- ofrece la posibilidad de eliminar la sobrecarga operativa gracias a un autoescalado en cuatro vías. A nivel de seguridad, propone el análisis de vulnerabilidades de las imágenes del contenedor y el encriptado de los datos. GKE, brinda dos modos operativos, en primer lugar el denominado 'estándar' que otorga un control absoluto sobre los nodos, así como la capacidad de ajustar y ejecutar cargas de trabajo administrativas personalizadas. Por otro lado, se encuentra el, 'Autopilot' que perfila como una solución automatizada y totalmente gestionada que se encarga de la infraestructura del clúster para que las empresas no se preocupen ni de su configuración ni de su supervisión. Dotado de plantillas y aplicaciones comerciales de código abierto creadas por Google para aumentar la productividad de los desarrolladores, los usuarios las pueden desplegar a golpe de clic no solo a través de la modalidad on-premise sino en nubes de terceros desde Google Cloud Marketplace. La inclusión de GKE Sandbox, una herramienta que añade una segunda capa de seguridad (y por lo tanto una protección mayor) entre cargas de trabajo en contenedores de GKE. La herramienta, que traslada aplicaciones desde máquinas virtuales a contenedores nativos, se basa en un sistema que extrae los elementos esenciales de las aplicaciones de las máquinas virtuales para insertarlos en contenedores de GKE o en clústeres de la plataforma Anthos sin ciertas capas de las máquinas virtuales (como el sistema operativo invitado) porque los contenedores no las necesitan.
HashiCorp Nomad	Además de organizar contenedores, cargas de trabajo heredadas y trabajos por lotes, Nomad permite que las compañías implementen y administren cualquier aplicación en cualquier nube. Se ejecuta como solución binaria única y admite una amplia gama de cargas de trabajo más allá de los contenedores, incluidos Java, Windows, entre otras. La plataforma permite a los equipos puedan migrar de forma incremental o almacenar en contenedores las aplicaciones existentes a su propio ritmo con un solo flujo de trabajo de implementación unificado	Se integra con las herramientas Vault y Consul para habilitar una solución completa y maximizar la flexibilidad operativa, pudiendo escalar a miles de nodos en un solo clúster e implementarse tanto en centros de datos privados como múltiples nubes. Las aplicaciones no contenerizadas, se orquestan con el objetivo de implementar y administrar aplicaciones 'heredadas' sin necesidad de reescribir o refactorizar. asimismo, se encuentra el escalador automático de Nomad (o Autoscaler) aprovisiona automáticamente a los clientes solo cuando se pone en cola un trabajo por lotes y los 'desmantela' una vez que se completa este trabajo; se ahorra así tiempo y dinero, ya que no hay necesidad de intervención manual. Entre sus usos ofrece una orquestación de contenedores simples para que las organizaciones implementen, administren y escalen contenedores en producción con facilidad Las organizaciones empresariales la pueden ejecutar como un servicio nativo de Windows gracias al sistema de proceso Service Control Manager y al administrador de servicios Non-Sucking Service Manager, como también por sus siglas NSSM.

Nota. Autoría Propia, basada en información del sitio web: Byte20-autora de la información, Regina de Miguel, (Regina de Miguel, 2022)

Docker Compose

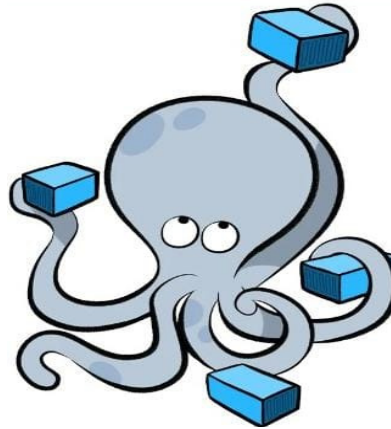
Docker Compose es una herramienta para definir y ejecutar aplicaciones de múltiples contenedores. Es la clave para desbloquear una experiencia de desarrollo e implementación optimizada y eficiente, además este simplifica el control de toda su pila de aplicaciones, facilitando la administración de servicios, redes y volúmenes en un archivo de configuración YAML único y comprensible, para luego con un solo comando, crea e inicia todos los servicios desde su archivo de configuración.

Trabajar con Docker Compose es posible en múltiples

ambientes, como la producción, puesta en escena, desarrollo, pruebas y flujos de trabajo de CI (integración continua), y también tiene comandos para gestionar todo el ciclo de vida de su aplicación:

- Iniciar, detener y reconstruir servicios
- Ver el estado de los servicios en ejecución
- Transmite la salida del registro de los servicios en ejecución
- Ejecutar un comando único en un servicio. (Dockers.docs, 2024)

Figura 3. Representación Gráfica del Archivo YAML



Nota. La imagen fue sustraída del sitio web Docker. docs Docker Compose Overview, 2024)

Tabla 4. Diferencias entre un Docker y Docker Compose

Máquina Virtual:	Contenedores
• Necesidad de un sistema operativo para cada servicio • Inicio y apagado lentos • Consume toda la RAM asignada a la MV • Red de transferencia de datos a través de forma de hipervisor. cuellos de botella • Comparta archivos con el Host a través del Hipervisor. Directo el acceso no es posible	• Comparte los componentes del sistema del host; no es Necesita un sistema operativo para cada servicio. • Inicio y apagado rápidos • Comparta archivos con el host mediante Copia segura (SCP) • Crea un puente de red para cada servicio que permite utilizar todo el ancho de banda. • Consume solo la RAM necesaria del servicio.

Nota. la información se retomó del sitio web (Docker Compose Overview, 2024)

Docker Compose es más adecuado para entornos de desarrollo y pruebas, donde se requiere una configuración rápida y sencilla de múltiples contenedores. Proporciona una experiencia de desarrollo fluida al permitirte definir y gestionar fácilmente un entorno local de desarrollo con varios servicios.

Sin embargo, para producción a mayor escala o de mayor volumen no es muy recomendable ocupar **Docker Compose**, empero, **Kubernetes** sería más apto ocuparlo, ya que es una plataforma más robusta y escalable diseñada para desplegar y gestionar aplicaciones en producción a gran escala, y ofrece características avanzadas de orquestación, como la gestión de clústeres, el balanceo de carga y la auto escalabilidad, lo que lo hace ideal para entornos de producción complejos y de alto rendimiento.

Igualmente es útil cuando necesitas configurar un entorno de desarrollo que involucra múltiples servicios o contenedores, y definir la estructura de tu aplicación y sus dependencias en un solo archivo YAML, simplifica la creación y gestión de tu entorno de desarrollo; y apesar de que se ocupe Docker Compose se forma individual, se obtienen beneficios como la capacidad de definir volúmenes y redes personalizadas, la posibilidad de escalar servicios y la facilidad de compartir y replicar tu entorno de desarrollo con otros miembros del equipo. Docker Compose ofrece una solución más completa y eficiente para configurar y orquestar aplicaciones multi-contenedor en tu entorno de desarrollo. (Dockers.docs, 2024)

Redes y Almacenamientos

Las redes en Docker son una herramienta que se encarga de definir cómo se comunicarán los contenedores de la plataforma entre sí, siendo el sistema de los contenedores Docker, y cuenta con distintos tipos de redes que son utilizadas para cumplir una gran variedad de labores y objetivos, asimismo tienen la posibilidad de interconectarse usando los llamados network drivers.

Tipos de Redes de Docker

Bridge: es la red estándar y network driver por defecto, es decir, se crea una vez se inicia la plataforma de Docker y sus contenedores se conectan a ella, a me-

nos que el usuario indique específicamente lo contrario, y usualmente se utilizan cuando las aplicaciones del cliente se ejecutan en contenedores independientes entre sí, requiriendo una comunicación aplicando aquellos a los containers que se ejecutan en el mismo host del demonio de Docker. Cabe resaltar que cada uno de estos contenedores cuenta con su propia red, que es, además, independiente de la red del host donde se encuentran ubicados.

Host: El controlador de redes en Docker denominado Host, se encarga de eliminar el aislamiento que pueda existir entre un contenedor de la plataforma y el host, igualmente cuando la red se encuentra en modo host, tiene la posibilidad de utilizarse con el objetivo de mejorar el rendimiento, asimismo este controlador de redes docker funciona en los momentos donde un container requiera controlar una gran diversidad de puertos, debido a que no necesita realizar el proceso de traducción de direcciones de red, ya que no se crea un proxy de usuario para cada uno de los puerto, utilizando el mismo puerto IP del servidor real que tenga el usuario; sin embargo, solo puede utilizarse en los hosts propios del sistema operativo Linux y no presenta compatibilidad con recursos como Docker Desktop versión MAC ni Windows, ni tampoco es compatible con Docker EE para los servidores del sistema operativo de Windows.

Overlay: Son las redes Docker que su función es crear una red distribuida entre distintos hosts del daemon de Docker, permitiendo que se realice la conexión de diversos contenedores en diferentes nodos del sistema, contribuyendo a la comunicación de los llamados **servicios de enjambre o swarm services** entre sí y con un contenedor de tipo independiente, o bien entre dos contenedores independientes que se encuentren en diferentes daemon de la plataforma. permitiendo que se pueda eliminar la necesidad de llevar a cabo un enrutamiento a nivel de sistema operativo entre los contenedores de Docker.

Ipvlan: es una red de contenedores que se encarga de ofrecer a los usuarios una gestión y control total sobre las direcciones IPv4 e IPv6, siendo un controlador Ipvlan, proporcionando a los operadores el total manejo del etiquetado de tipo VLAN de capa 2, e incluso de elementos como el enrutamiento Ipvlan L3 para aque-

llos usuarios que quieran realizar procesos de integración de la red subyacente.

Macvlan: Es una de las redes en Docker que se utiliza para asignar una dirección MAC a un contenedor determinado, haciendo que aparezca como si fuera un dispositivo físico en su red, dando al daemon Docker que permite enrutar el tráfico hacia los contenedores a través de sus direcciones MAC, y controlar las redes en Docker, teniendo una gran utilidad para el manejo de aplicaciones que han sido heredadas y que esperan conectarse de forma directa con la red física, en lugar de ser enrutadas mediante la pila de red del host de la plataforma.

None: esta opción de red de contenedores es la encargada de inhabilitar todas las redes de la plataforma de contenedores, y usualmente se emplea en conjunto con un controlador de red personalizado por el usuario. (KEEPCODING, 2022)

Tabla 5. Comandos y Funciones de las Redes:

- **Docker network create:** es el encargado de crear una nueva red en el sistema.
- **Docker network ls:** ofrece un listado de las redes que tiene el usuario.
- **Docker network rm:** permite borrar una o más redes.
- **Docker network connect:** es el comando que permite el proceso de conexión de un contenedor a una red.
- **Docker network disconnect:** cumple con la función de desconectar un contenedor a la red.
- **Docker network inspect:** esta opción se utiliza para obtener información detallada acerca de una determinada red.

Nota. la información se sustrajo del sitio web KEEPCODIG-Tech School (KeepCoding, 2022)

Volúmenes de Dockers

Es un componente crucial del ecosistema que almacena y gestiona datos persistentes generados por contenedores efímeros, que permiten a los datos persistir incluso después de eliminar o actualizar un contenedor, por lo tanto los datos esenciales de la aplicación no se pierdan durante las operaciones rutinarias, estando desacoplados en el sistema de archivos del contenedor, por lo que se pueden hacer fácilmente co-

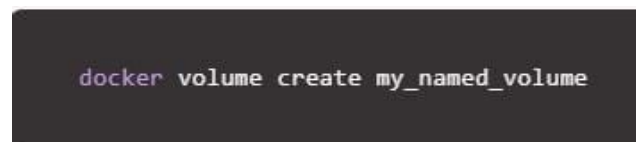
pias de seguridad de ellos, y compartirlos entre varios contenedores y migrarlos entre hosts.

La ventaja del uso de volúmenes en lugar de montajes vinculados (**sbind mounts**), son los montajes de directorios desde el sistema host a un contenedor, haciéndolo portátil, dándole la posibilidad de trasladar volúmenes rápidamente entre diferentes hosts o contenedores, pero debes vincular los montajes vinculados a un directorio específico en el sistema host, asignando una gestión de datos más flexible y eficaz en las aplicaciones basadas en contenedores.

Los volúmenes también son compatibles con varios controladores de almacenamiento, lo que te permite elegir la mejor solución de almacenamiento para tu caso de uso específico, y existen dos tipos distintos: nombrados y anónimos.

Volúmenes Nombrados: son los que tienen un nombre definido por el usuario, lo que facilita su identificación, gestión y uso compartido entre varios contenedores, crea y gestiona los Dockers en volúmenes nombrados y almacena sus datos en una ubicación específica del sistema host, y su ubicación suele estar dentro del directorio de instalación de Docker, bajo un ID único correspondiente al nombre del volumen, ofrecen mayor control y flexibilidad, ya que puedes referenciarlos y manipularlos fácilmente utilizando su identificador legible por humanos. (Marcia Ramos, 2023)

Figura 4. Comandos para Volúmenes Nombrados



Nota. La imagen se retomó del Sitio web: Marcia Ramos, (2023)

Volúmenes Anónimos: A diferencia del anterior, no tienen un nombre definido por el usuario, en su lugar, Docker los crea automáticamente cuando creas un contenedor y asigna un ID único al volumen, por lo general, es más difícil gestionar y almacenar volúmenes debido a que carecen de un identificador legible por humanos, y por su origen es habitual utilizar volúmenes anónimos para el almacenamiento temporal, además pueden aparecer si no especificas un volumen con nombre al crear un contenedor.

Figura 5. Comandos para Volúmenes Anónimos

Nota. La imagen se retomada del sitio Web: (Marcia Ramos, 2023)

CONCLUSIÓN

Finalmente, la investigación muestra las virtudes de la Tecnología Docker, su fácil administración e implementación en diferentes sistemas operativos, operando de manera ordenada y conjunta para la ejecución de su código; Simultáneamente, Docker demuestra su utilidad desde los contenedores para empaquetar y ejecutar aplicaciones de manera aislada, lo que permite optimizar la portabilidad y eficiencia, teniendo como herramientas Dockerfile, Docker Compose, Docker Hub, Contenedores, Volúmenes y Redes, mencionando delimitadamente sus utilidades básicas, ya que la temática es muy amplia como para únicamente quedarse en una nota técnica de investigación, instando que la presente investigación sirve como punta del iceberg para fomentar la aplicabilidad de las nociones básicas descriptivas hacia una fase de pruebas por parte de los lectores, investigadores interesados, estudiantes en sistemas informativos y público en general, porque la utilización de docker en los diferentes procesos de creación de los distintos tipos de sistemas y aplicativos informáticos representará en pocas décadas la ejecución de dichos sistemas más avanzados, complejos y provechosos para la humanidad.

REFERENCIAS

- ARSYS. (n.d.). Servidores Cloud. 30 días de prueba gratuitos. Arsys. Retrieved September 30, 2024, from <https://www.arsys.es/servidores/cloud>
- AWS. (n.d.). Contenedores de Docker | ¿Qué es Docker? | AWS. AWS. Retrieved September 30, 2024, from <https://aws.amazon.com/es/docker/>
- Curso de Docker [Gratis y Certificado]. (n.d.). Edutin Academy. Retrieved September 30, 2024, from <https://edutin.com/curso-de-docker-3986>
- Curso de Google Cloud Platform GCP [Gratis y Certificado]. (n.d.). Edutin Academy. Retrieved September 30, 2024, from <https://edutin.com/curso-de-google-cloud-platform-gcp>
- de Miguel, R. (2022, July 19). COMPARATIVA: 10 plataformas de gestión de contenedores. Revista Byte. Retrieved September 30, 2024, from <https://revistabyte.es/comparativa/comparativa-10-contenedores/>
- Docker. (n.d.). Install Docker Desktop on Linux. Docker Docs. Retrieved September 30, 2024, from <https://docs.docker.com/desktop/install/linux-install/>
- Docker. (n.d.). Install Docker Desktop on Mac. Docker Docs. Retrieved September 30, 2024, from <https://docs.docker.com/desktop/install/mac-install/>
- Docker. (n.d.). Install Docker Desktop on Windows. Docker Docs. Retrieved September 30, 2024, from <https://docs.docker.com/desktop/install/windows-install/>
- Docker Docs. (n.d.). Docker Compose. Docker Docs. Retrieved September 30, 2024, from <https://docs.docker.com/compose/>
- Docker Docs. (n.d.). Dockerfile reference. Docker Docs. Retrieved September 30, 2024, from <https://docs.docker.com/reference/dockerfile/>
- Fredrickson, P. (2019, Diciembre 25). Aprende Docker en 14 minutos. Aprende Docker en 14 minutos. <https://www.youtube.com/watch?v=6id-FknRIOp4&t=242s>
- Gutiérrez, M. (n.d.). Tutorial completo de Docker. Aprender Gratis. Retrieved September 30, 2024, from <https://aprendergratis.es/cursos-online/tutorial-completo-de-docker/>

- Herrera, L. (2017, Mayo 17). A performance evaluation between Docker container and Virtual Machines in cloud computing architectures [Google Translator] Pontificia Universidad Católica de Ecuador. A performance evaluation between Docker container and Virtual Machines in cloud computing architectures. <https://publicaciones.ucuenca.edu.ec/ojs/index.php/mas-kana/article/view/1457/pdf>
- Ingeniería de Sistemas Informáticos. (n.d.). Secretaría General. Retrieved September 30, 2024, from http://secretariageneral.ues.edu.sv/index.php?option=com_content&view=article&id=60&Itemid=94
- Lopez, A. (2021, Abril 11). ¿Qué es un Contenedor e Imagen en Docker? ¿Qué es un Contenedor e Imagen en Docker? <https://www.youtube.com/watch?v=FAJ1o3hb35s&t=85s>
- Nuestra Universidad. (n.d.). Universidad de El Salvador. Retrieved September 30, 2024, from <https://www.ues.edu.sv/nuestra-universidad>
- Ramos, M. (2023, September 5). Utiliza Volúmenes para Gestionar Datos Persistentes con Docker Compose. Kinsta. Retrieved September 30, 2024, from <https://kinsta.com/es/blog/volumenes-docker-compose/>
- Revisión Sistemática de la Literatura Sobre Tecnologías de Docker [Tesis de Magister, En Tecnología Analítica, Universidad de Colombia]. (2022). Revisión Sistemática de la Literatura Sobre Tecnologías de Docker. <https://repositorio.unal.edu.co/bitstream/handle/unal/81460/1001359126.2022.pdf?sequence=1&isAllowed=y>
- Rodríguez, F. (2023, January 27). ¿Qué son las redes en Docker? KeepCoding. Retrieved September 30, 2024, from <https://keepcoding.io/blog/que-son-las-redes-en-docker/>
- Sánchez Hernández, J. J. (2022). Aprender Docker Un Enfoque Práctico, N°1. Aprender Docker Un Enfoque Práctico, N°1. <https://cbues.bibliotecasdigitales.com/read/9788426734488/index>